**Computer Network Reading Assignment #1**

컴퓨터공학부

2009003299

남혁주

1. **introduction**

이 논문은 컴퓨터 네트워크에 사용되는 TCP/IP 프로토콜이 왜 만들어 지게 되었고 설계의 목적과 구현된 방법 사이의 차이에 대해서 쓰여졌다. 1970년대 초에 packet switched networking에 적합한 프로토콜이 개발되었다. 그것이 Internet Protocol(IP) 와 Transmission Control Protocol(TCP)이다. TCP/IP 설계에 대한 철학은 첫 번째 제안서로부터 많이 진화하고 발전해왔다. Internet Architecture의 원래 목적을 먼저 확인해보고 제안서에 있는 목표들과 프로토콜의 중요한 특징을 비교해보면서 생각해보려고 한다.

1. **Fundamental Goal**

DARPA Internet Architecture의 가장 중요한 목적은 연결되어있는 네트워크들을 다중화 기능을 하는 효율적인 기술을 개발하는 것이었다. 그래서 이 Architecture의 원래의 목적은 이미 있던 네트워크들을 ARPA packet radio network를 이용해서 ARPANET에 연결하여 사용하는 것이었다. 이렇게 네트워크들을 연결하여 사용하는 것은 높은 성능을 내는 것이 확인 되었다. 그래서 네트워크들을 연결하는 것이 필요하다고 느끼기 시작했다. 그 후 다중화 기능을 사용하기 위한 기술로 packet switching이 선택되었다. 그리고 packet switching은 Internet Architecture의 기본적인 구성요소로 받아들이게 되었다. 마지막으로 이 기본적인 목적에는 특정한 기술이 필요한데 그 기술은 packet을 저장하거나 보내는 기술이다. 그리고 이 Internet packet switch에 이용되는 layer를 gateway라고 부르기도 한다.

인터넷의 기본적인 구조를 정리하면 packet switch communication은 패킷을 보내고 받는 기능을 하는 packet communication processor (gateway)를 사용해서 많은 다양한 네트워크들을 연결하는 것이다.

1. **Second Level Goals**

Internet Architecture에는 다양한 목표들이 있는데 1. 인터넷 통신은 네트워크나 게이트웨이가 없어도 계속되어야 한다 2. 인터넷은 다양한 통신 서비스를 지원해야 한다. 3. 인터넷 구조는 다양한 네트워크를 수용해야 한다. 4. 인터넷 구조는 자원을 분산시키는 것을 허용해야 한다. 5. 인터넷 구조는 비용을 효율적으로 해야 한다. 6. 인터넷 구조는 적은 노력으로 host 연결을 허용해야 한다. 7. 인터넷 구조에 사용되는 자원은 책임이 있다. 이 목표들은 중요한 순서대로 나열되어 있는 것이다. 여기서 중요한 것은 시대가 변하면서 인터넷 구조가 지향하는 목표들이 변한다는 것이다. 전쟁시대에 중요하게 생각했던 목표들이 있었지만 시대가 변하면서 목표들이 추가되고 또 중요도도 같이 변할 수 있다.

1. **Survivability in the Face of Failure**

이 리스트 중에서 가장 중요한 목표는 네트워크나 게이트웨이가 실패하더라도 통신 서비스를 유지하는 것이다. 이 목표를 얻기 위해서는 전송중인 상태를 보호해야만 한다. 예를 들어 state의 정보를 저장해서 얼마나 보내고 받아야 하는 지에 대한 정보를 알고 있으면 서비스를 리셋 할 필요 없이 통신을 유지할 수 있다. 하지만 낮은 layer에서 이 정보를 잃는 다면 동기화가 불가능 하고 문제가 발생한다. 낮은 layer에서 문제가 발생하지 않는다면 이 구조는 패킷을 잃는 것에 대해서 보호되어 있다고 할 수 있다. 다른 인터넷 구조에서는 이 state가 네트워크의 패킷 스위칭 노드에 바로 저장이 되어있다. 이 경우에는 패킷을 잃는 것에 대비해서 패킷들을 복사해 둬야 한다. 그래서 이 문제에 대한 대안으로 fate-sharing을 제안했는데 이 기법은 인터넷의 끝부분에서 이 정보들을 모으는 것이다. fate-sharing은 entity 자체를 잃는다고 해도 처리할 수 있다. fate-sharing에는 복사하는 것보다 2가지의 중요한 장점이 있는데 (1) 복사는 1개의 실패만 처리할 수 있지만 fate-sharing은 다양한 실패를 처리할 수 있고 (2) 복사보다 쉽게 구현할 수 있다. fate-sharing이 survivability와 연결된 2가지 결론을 지으면 (1) 스위칭 노드나 게이트웨이가 정보를 저장할 필요가 없고 (2) 호스트에서 관리하는 것이 더 신뢰성이 높다.

1. **Types of Service**

2번째 목표는 다양한 서비스를 지원해야 하는 것인데 이 부분에서 TCP를 사용한다. 초기의 TCP 컨셉은 다양한 서비스를 지원하는 것이었지만 모든 부분에 지원하는 것이 너무 어려운 상황이었다. 그리고 TCP로는 불가능한 네트워크 연결 분야도 있었다. 그래서 나온 것이 TCP와 IP를 탄생시켰다. TCP가 모든 부분을 지원하는 것 대신에 IP에서 다양한 서비스를 지원하고 TCP는 하나의 서비스만 실행하는 것이다. 이 때 building block을 사용해서 TCP와 IP를 연결하는데 이 것을 datagram이라고 이름 지었다. 신뢰성은 완벽하게 보증하지는 못하지만 그래도 가장 효과적인 기술이었다. 그리고 UDP가 개발되었는데 이 것은 application layer interface를 제공하기 위해서 만들어졌다. TCP/IP가 다양한 서비스를 지원한다는 것을 위반하지 않기 위해 datagram을 사용하는 방법을 선택한 것이다. 그래서 인터넷을 하기 위해서는 datagram으로 호스트와 게이트웨이를 연결하는 기술로 나아가게 되었다.

1. **Varieties of Networks**

다양한 네트워크의 기술을 연결하는 것을 가능하게 하는 기술이 가장 중요하다. 인터넷 구조는 이런 면에서 아주 성공적으로 목표를 달성했다. 다양한 네트워크 기술들이 있는데 이 기술들이 연결되어 인터넷 구조를 지원하고 있다.

1. **Other Goals**

위에서 말한 3가지의 목표들 이외의 목표들은 중요성이 낮다. 그래서 효율적인 방법을 아직 찾지 못하거나 완벽하게 만들어지지 않았다.

1. **Architecture and Implementation**

인터넷 구조로 연결된 특정한 네트워크와 게이트웨이와 호스트의 집합을 realization이라고 한다. 이 realization은 다양한 곳에 다양하게 설정되어 있다. 다양한 realization이 설정되어 있는데 이 것의 성능을 측정하는 것은 매우 어려운 일이다. protocol verifier와 simulator를 사용할 수 있지만 그래도 한계가 있다.

1. **Datagrams**

인터넷의 기본적인 구조적 특징은 Datagram의 사용을 본질로서 사용하는 것이다. Datagrams사용하면 연결 상태에 대한 정보를 스위칭 노드에서 저장할 필요가 없다. 그리고 다양한 종류의 서비스를 구현할 수 있다. 마지막으로 Datagram은 가장 적은 서비스에 대한 가정을 요구한다.

1. **TCP**

원래 ARPANET의 host-to-host 프로토콜은 플로우 컨트롤을 바이트와 패킷 2가지를 모두 사용했다. 하지만 이 방법은 복잡해서 하나의 방법으로 통일하기를 원했다. 그리고 패킷보다는 바이트로 하는 것이 선택되었다. 첫 번째 이유로 제어 정보를 바이트의 순서 공간에 넣는 것을 허용하기 위해서이다. 이 것으로 데이터를 acknowledge로 사용할 수 있게 되었다. 두 번째 이유는 TCP 작은 패킷 사이즈로 줄이고 싶다면 패킷을 잘라서 사용할 수 있게 하기 위해서이다. 세 번째 이유는 패킷을 재전송할 필요가 있을 때 작은 패킷들을 모아서 패킷 덩어리로 보내기 위해서이다. 다시 생각해보면 TCP는 다양한 서비스를 지원해야 한다. 그래서 원래의 ARPANET 프로토콜은 2가지의 데이터 형태를 다 사용할 수 있다.

1. **Conclusion**

Datagram은 가장 중요한 인터넷의 목표를 잘 해결한다. 하지만 자원 관리나 책임에 관한 부분에서는 해결하기가 어렵다.